National College of Ireland

BS(Hon's) in Computing

Software Development

2021/2022

Luke Kavanagh

X17520686@student.ncirl.ie

T – Exchange: Ticket Exchanging System

Technical report

Contents

1.0 Introduction 4

1.1 Background 4

1.2 Aims 5

1.2.1 Gives Users Alternative option 6

1.3 Technology 6

JavaScript – React. 6

Nodejs 6

Express.js 6

MongoDB - Mongoose 6

Redux – 6

1.4 Structure 7

2.0 System 8

2.1 Requirements 8

2.1.1 . Functional Requirements 8

2.1111 Use Case Diagram 8

2.1.1.1.2 Requirement 1: Describing the process 8

2.1.1.1.3 Description and priority 8

2.1.1.1.4 Use Case 8

Requirement 2 – Login System 10

Flow description 11

Requirement 3 – Exchanging/Getting Tickets 12

2.1.2. Data Requirements 13

2.1.3 User Requirement 13

2.1.3.1 Security 13

2.1.3.2 GDPR – Data Protection 13

2.1.4 Usability Requirements 14

2.1.4.1 Navigation 14

2.1.4.2 User friendly 14

2.1.4.3 Design 14

2.2 Design and Architecture 15

2.2.1 Class Diagram 15

2.3 Implementation – 15
Database 15
2.3.2 Back-end Development 17
2.3.3 Front–end Development 19
2.4 Graphical User Interface (GUI) 21
Main Page Draft 21
2.5 Testing 21
Unit testing 21
Jest 21
3.0 Conclusions 22
4.0 Further Development and Research 22
5.0 References 23
6.0 Appendices 24
6.1 Project Proposal 24
1.0 Objective 25
2.0 Backgrounds 26
3.0 State of the Art 26
4.0 Technical Approach 27
5.0 Technical Details 28
6.0 Special Resources Required 28
7.0 Project Plan 28
.... 31
6.2 Reflective Journals 32
6.3 Invention Disclosure Form 46

Executive Summary

The main purpose of this project is too address a situation that is currently ongoing in the area of ticket purchasing with a goal and view to improve the services that are being provided as we speak. The project itself is based on a ticket exchanging system that will ensure that a customer will have the ability to purchase a ticket for a concert or a different event from another person and will not be overcharged by anymore then what the original asking price was for the ticket. We have seen for years the ongoing struggle of customers trying to purchase tickets for events only to find them and 3rd party websites for triple the asking price. With new laws making more difficult for people to sell tickets at these events we are trying to make it easier for not only the customer but for the person selling the Tickets. We have seen that some of the new penalties that are being put into place include €100,000 or even up too two years in prison for anybody who is caught defying the sale of tickets though the bill of 2021 Cultural, Entertainment, Recreational and Sporting Events), this was approved by the President Michael D. Higgins.

T-Exchange will allow for customers not only benefit by purchasing tickets for their chosen event but easier in that they do not have to pay more or then the original asking price. The application will be able to improve and assist both sides of the parties in that people selling the tickets won't have a massive gain and can get their money back and the customer.

As I mentioned before that application is designed to benefit the user and make sure there isn't any issues that might occur. Throughout the process of doing the project report I will talk about how I carried out doing the project to its extent.

1.0 Introduction

1.1 Background

The key goal for this project is too focus on the important issue surrounding that the area of selling and purchasing tickets. We saw in July of 2021 that the Irish government made ticket touting illegal and it has definitely made some good benefits form it. (Depeartment of Enterprice, 2021) however we did notice it did have certain flaws. Some events companies are preventing people from gaining access to an event if they are not the ticket holder or the name of the person who purchased the ticket in the first place. This will leave people not only disappointed by not getting to attend the event but also they are losing money in this event.

With the increasing demand for people to return back to events and go out socializing during the Covid 19 Global Pandemic gets bigger we are still looking at the safe way of making this happen. Even when we look at how people are not using paper tickets anymore and are starting to use QR codes to scan into the events.

Ticket swap: Ticket Swap is becoming popular in that people are purchasing there tickets online. Its website states that it has over 1.500,000 who have sold their tickets successfully. It also benefits that it has the following

- User friendly interface
- Easy of navigator to use
- Login system

The main reason why I am looking at possibly implementing a QR code system is that the information will be there and it shows evidence of the purchase. It will show that the customer who has purchased the tickets have all the details and will be able to gain access to the event.

The project will allow for the user to post up any event that has

1.2 Aims

The main aim for this project is to build a Desktop Application Ticket exchanging system. This software will primarily focus on the desktop application. My main target audience or customers will be mainly people who are interesting in attending events or who are looking to sell a ticket they bought, The application will be able to help people from both a purchasing side and from a point of making their money back from selling the tickets. Some of the features that will be provides include

- Safe and secure login system
- Users can create an account
- Allow them to see if they can purchase ticket a certain events
- Allow them to enter there details
- Give a purchase/sell ticket option to user
- Enter details for an event they want to see
- Purchase as many tickets you want

This application will be able to help out a lot of people from both sides. With these new laws coming into play and events ensuring that only the ticketholder can gain access to the event it will benefit a lot of people in the long term. As mentioned the application on their desktops will be able to perform all the tasks that have been set out for the project. The process of purchasing a ticket through the application will take a matter of minutes.

1.2.1 Gives Users Alternative option

T- Exchange gives the customers and the users more options to choose from the application, With the new regulations in place it will make it more difficult for users to sell tickets especially just after the rules regarding the Covid 19 pandemic. The application is safe and makes it easier for the user to exchange there tickets without any problems.

1.3 Technology

In the development and research for this project I looked at most of the possible outcomes in which tools to select. After a tough choice a decided in the end to select, JavaScript, React.js, NodeJS, MongoDB and express.js and Heroku for hosting. Below is a full detailed plan of the technologies being used

JavaScript – React. React has become one of the most popular JavaScript frameworks with most being using it as their main. This will be used for the main functions within the application and will help design some of the main pages within the application. It makes it easier to develop creative user interfaces within the system.

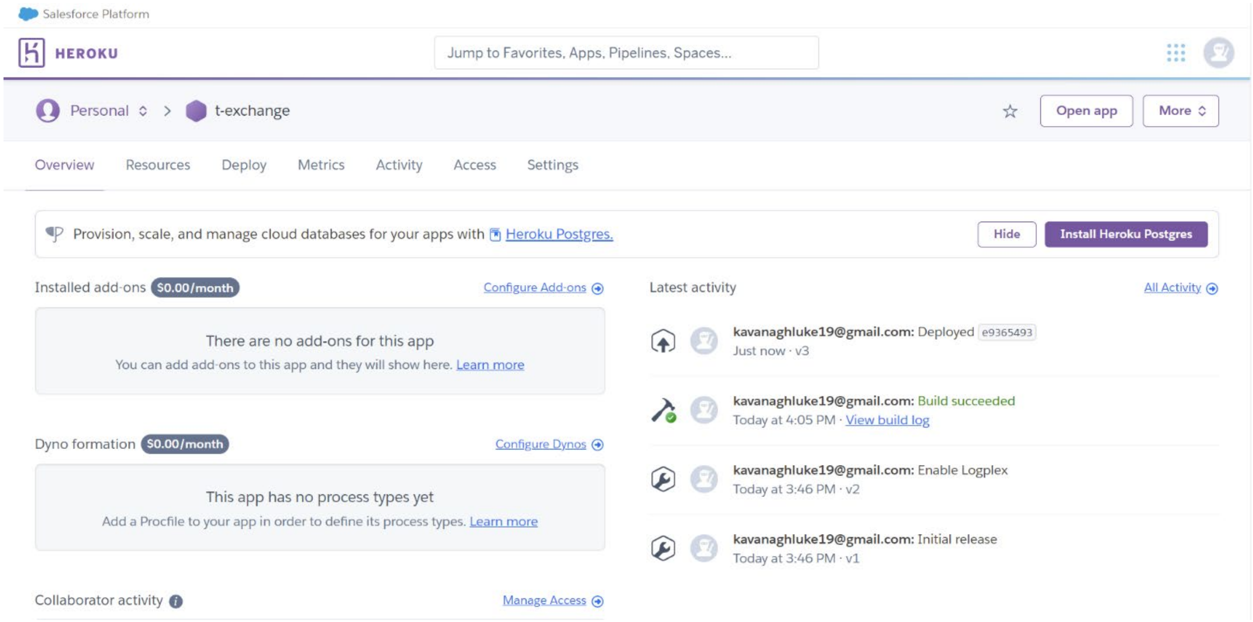
Nodejs **–** Node is used for most websites and is used for Backend API services. It has the ability to create, delete, open or read files on any server. It can also modify data on a database.

Express.js **– fr**om doing research into the project I came across express.js, I have never used it before so I quickly did my research on it. It's a node.js framework so it will come in handy when needed in the project

MongoDB - Mongoose – Mongo is becoming one of the most reliable databases today, I used it last year on my internship and wanted to use it again for my final year project. MongoDB will be used to store the data that is in the application, when the user will post up a ticket event then the data will be stored in a cluster on MongoDB.

Redux – Redux will be used to help React.js improve the relationship between certain objects and to facilitate the management of objects within the system.

Heroku – Heroku was implemented for the deployment of the project. It will host our application which will allow for me to test it when it goes live.



Provision, scale, and manage cloud databases for your apps with Heroku Postgres.

Installed add-ons $0.00/month

There are no add-ons for this app
You can add add-ons to this app and they will show here. Learn more

Dyno formation $0.00/month

This app has no process types yet
Add a Procfile to your app in order to define its process types. Learn more

Collaborator activity

Latest activity

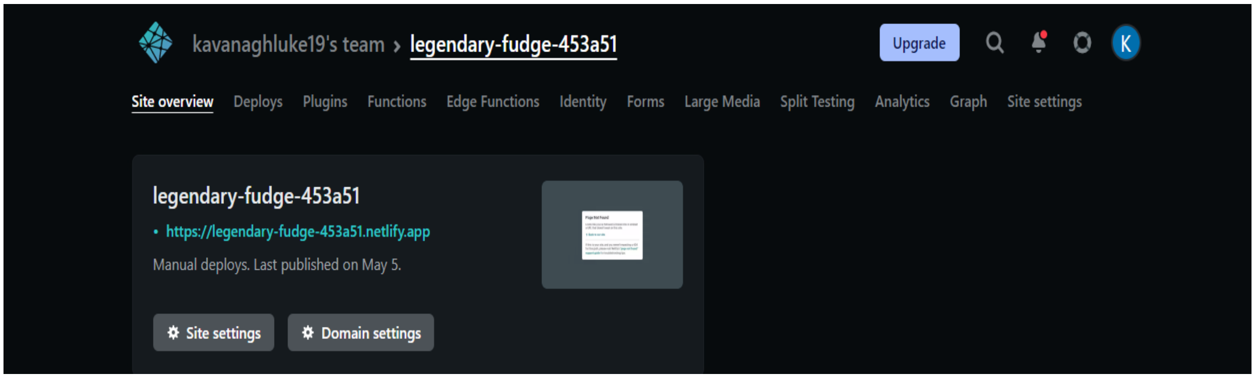
kavanaghluke19@gmail.com: Deployed e9365493
Just now · v3

kavanaghluke19@gmail.com: Build succeeded
Today at 4:05 PM · View build log

kavanaghluke19@gmail.com: Enable Logplex
Today at 3:46 PM · v2

kavanaghluke19@gmail.com: Initial release
Today at 3:46 PM · v1

Netlify – Netlify is used as a server that is good for hosting. It good to use especially for front end development. We are using this too so that it separates both Heroku and Netlify for the front -end.



legendary-fudge-453a51
https://legendary-fudge-453a51.netlify.app
Manual deploys. Last published on May 5.

1.4 Structure

The structure of the document is done through the following, The requirements section will include both functional which will include UML diagrams and the use cases and the non-functional requirements. The Architecture and design will include the details about the entire system, its architecture design and the key main components. Such as Hardware, Software and System Architecture. This will follow by the GUI, Analysis and Design, the testing Evaluation with finally the Conclusion and the future development within the project.

Structure will include

- Functional Requirements – it will show what the application will set out to do to be functional
- Design and Architecture – Shows a quick diagram of how the project will work in the project.
- Implementation – Provides details of the application from the full Stack Development side to the Database.
- Graphical User Interface – Sample User interface that was planned in the early stages.
- Testing and Evaluation – Gives details of how the testing was to be carried out in the application.

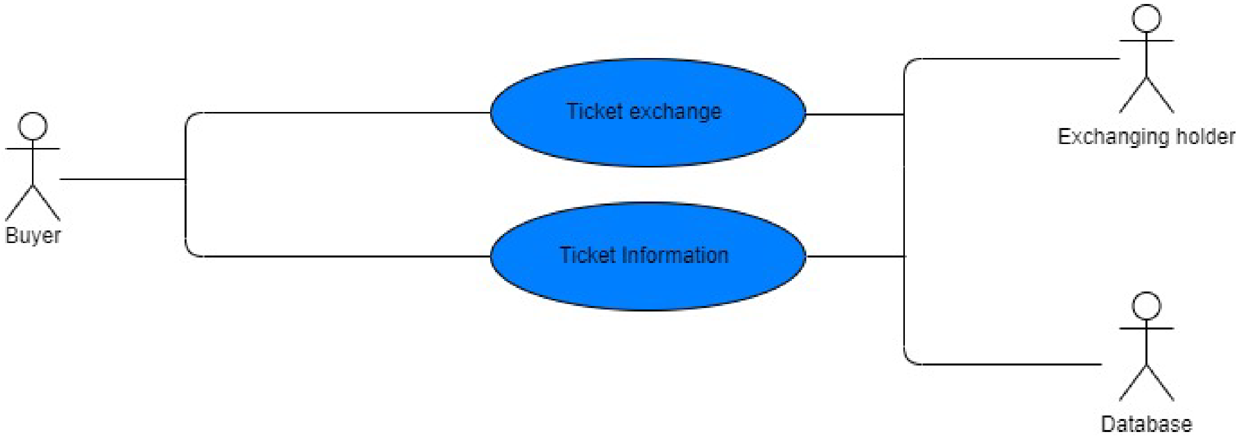
2.0 System

2.1 Requirements

2.1.1 . Functional Requirements

The functional requirements of T-Exchange Ticket System will explain the requirements that are needed to overcome the project and get the results.

2.1111 Use Case Diagram



2.1.1.1.2 Requirement 1: Describing the process

This use case describes the process of the entire project

2.1.1.1.3 Description and priority

The is the main process of the body and project, The use case explains the process of how when the ticket information is exchanged and is saved within the database

2.1.1.1.4 Use Case

Unique ID: Ticket Exchange

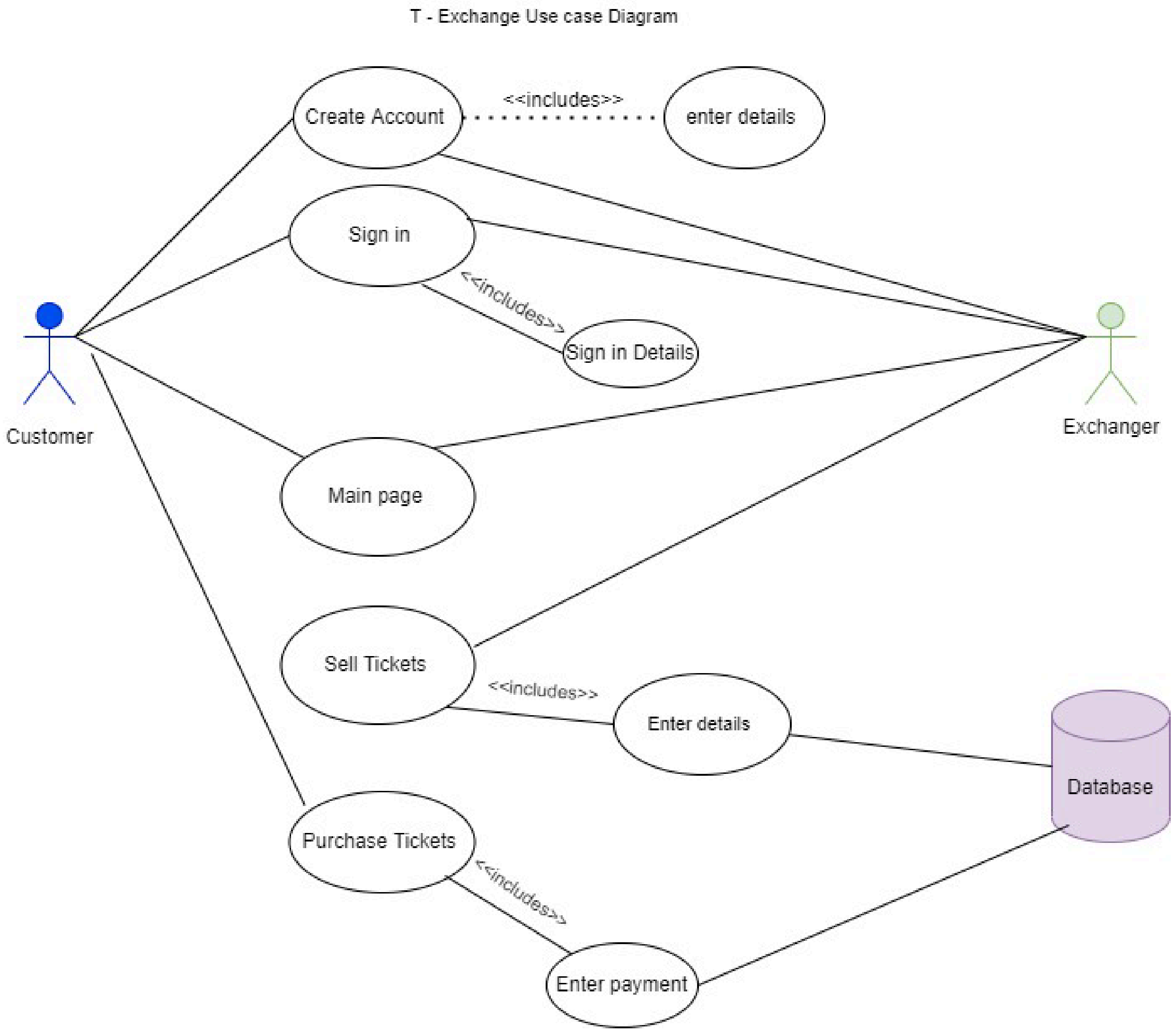
Scope

The Scope of the use case is for the customer/buyer to purchase the ticket from the exchanging holder

Description

The use case describes how the buy the ticket off the person looking to exchange it, it will be exchanged through a payment and given a code to work.

Use Case Diagram – Overall Project



Flow Description

Preconditions

- The internet needs to be connected
- The Application needs to be turned on

- The application can be defined by the system
- The exchanger has his ticket details uploaded already
- The buyer in the use case is looking to purchase a ticket

Activation

The use case starts when the user logs into the log in page

Main Flow

- The User opens up the page
- They then are asked to create a account
- They need to enter their full name, email address and create a password
- Email needs to contain'@' to progress
- Password need to be more than 8 characters and contain a number and symbol
- They must be both uppercase and lowercase
- If unsuccessful, the user is asked to try again
- The user enters the main page
- The select there open to either Purchase tickets or Sell Tickets
- They then must enter the details to progress
- The Ticket price cannot exceed the amount of the original face value
- The buyer must enter their email and code
- The details will be stored in a local database
- The exchanger will receive their money when all is successful
- If all details don't match it will send an error message
- A message will appear saying you have purchased this ticket
- Other details will appear like price and code
- The user will then return to the main page

Alternative flow

A1<No access to internet**>**

5. The internet isn't working

Requirement 2 – Login System

Description and priority

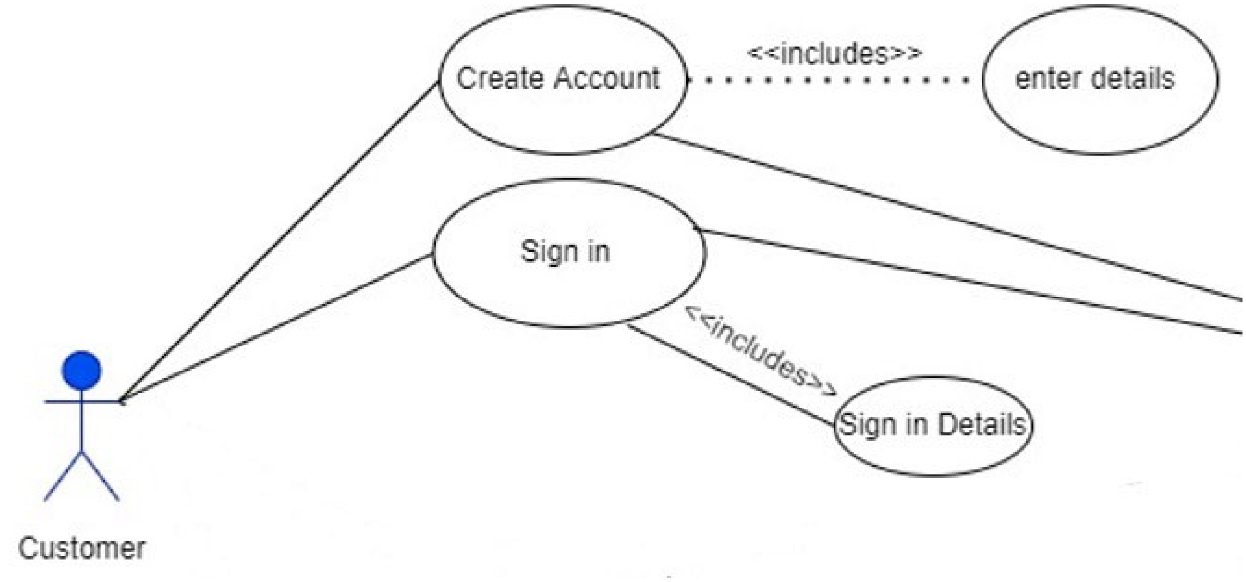
For T-Exchange to work successfully each user has to log in or sign up in order to preform tasks within the system.

Use case

Scope

Description

The user needs to Log in to navigate to the ain page.



Flow description

Precondition: user must get to the main page before signing in

Activation flow : The use case starts when the user goes into login to the system

Main flow

1. The user selects the sign up button
2. The user is presented with the sign in page
3. The user enter in all the details correctly
4. The user is then brought to the main page

Alternate flow

1. The user clicks the Login button
2. The user enters the details
3. The details are incorrect
4. The details are re-entered and correct
5. The user is brought to the main page

Exceptional Flow

1. The user gets to the login page
2. The details entered are incorrect
3. The user leaves the login page

Termination: the information is then stored in the system and is directed to the main page.

Post Condition: The user is now able to view the main page and see what tickets are available or can add an event to be shared with other users.

Requirement 3 – Exchanging/Getting Tickets

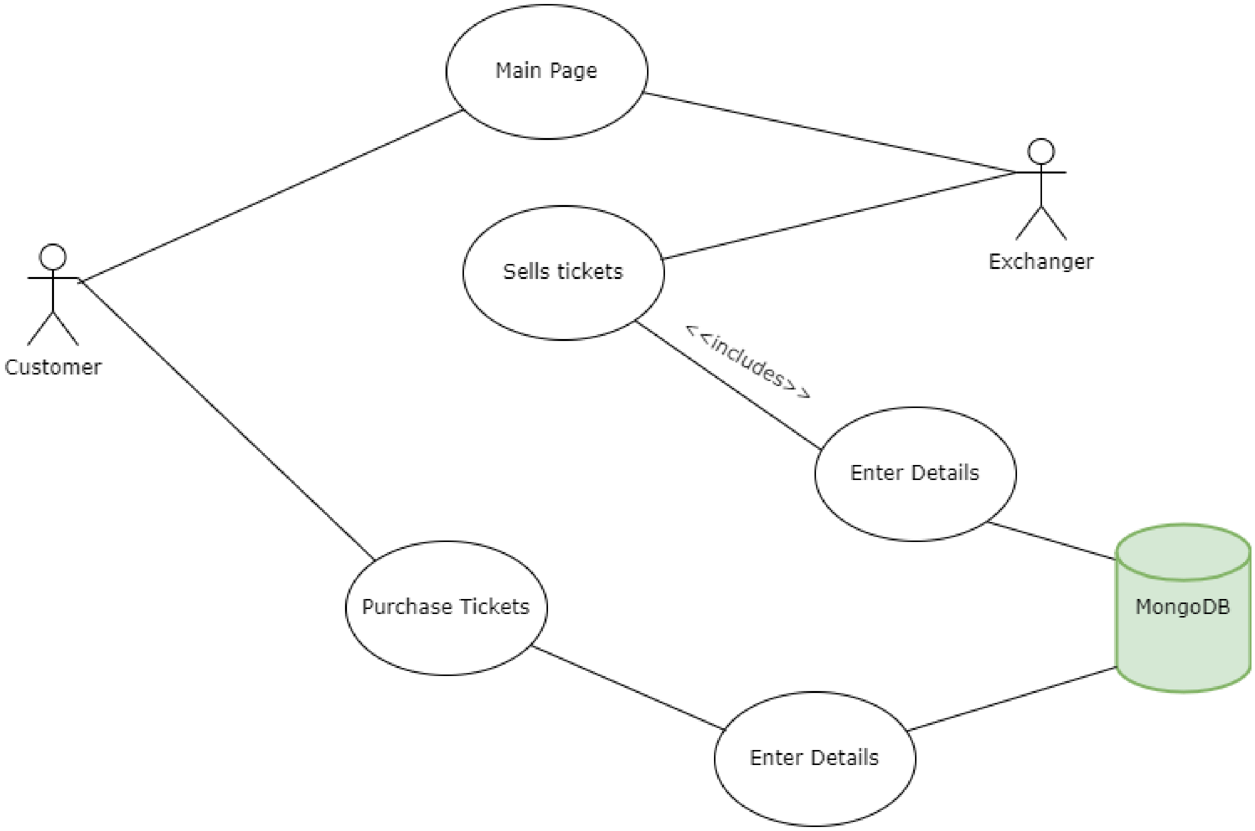
Scope

The scope of this use case allows for the user to exchange their ticket

Description

This use case describes the actors involved in the exchanging process

Use Case Diagram



Flow Description

Precondition

The System is in wait stage until the user begins to enter details.

Main flow

1. The use case starts when the when the user selects enter a event
2. The user will then enter all the details that it needs
3. The user enters the details
4. The event is created and posted on the main page for other users to see

Alternative flow

1. The system posts the event onto the main page
2. The event will be viewed by other users
3. The event is either selected or denied

Termination

The system redirects the user to the main page

Post condition

The system goes into a wait state.

2.1.2. Data Requirements

T – Exchange system and user interface were all designed so that the users to have a straight forward experience when using the application. Looking through resources for each of the technologies on how this project can be implemented together.

The languages used for the include the JavaScript front-end technologies including React.js and with some Redux. These are used to process, analysis and to display the data that is within the application. Node.js and Express are also used to help retrieve the data from the MongoDB atlas with the help of HTTP protocols.

2.1.3 User Requirement

The User will need to have a certain understanding how do you use internet or applications. Could have certain access to social media, Have access to a smart phone to preform searches on your phone or PC.

2.1.3.1 Security

The security of any system is a key requirement. It is also important once there is possession of users details that they will all be safely handled, The application would also try prevent any cyber security attacks that could happen with the possibility of having information leaked.

2.1.3.2 GDPR – Data Protection

GDPR is one of the highest requirements that a application needs to have much attention too especially if there user data being processed, like we mentioned in security peoples information. The users will have the ability within the system to change details or delete them as they progress within the application. In Irish law as of 25 May 2018 it shows that the default that the majority of personal data processing will apply by the GDPR by then which will be set out in the data protection Act 2018 (Anon., 2018).

2.1.4 Usability Requirements

The application is very easy and straightforward to use. The color pattern will be easy for people to use, Green and red colors will be used throughout to signal to proceed or not.

The text within the application will be readable for the users and won't make it overcomplicated for the user to access certain features.

2.1.4.1 Navigation

As a developer it is our job to not over complicate a website or a application, With T- Exchange it keep everything fast and loading especially the fact that the website is developed through React components which improved both the look and the feel of the website.

2.1.4.2 User friendly

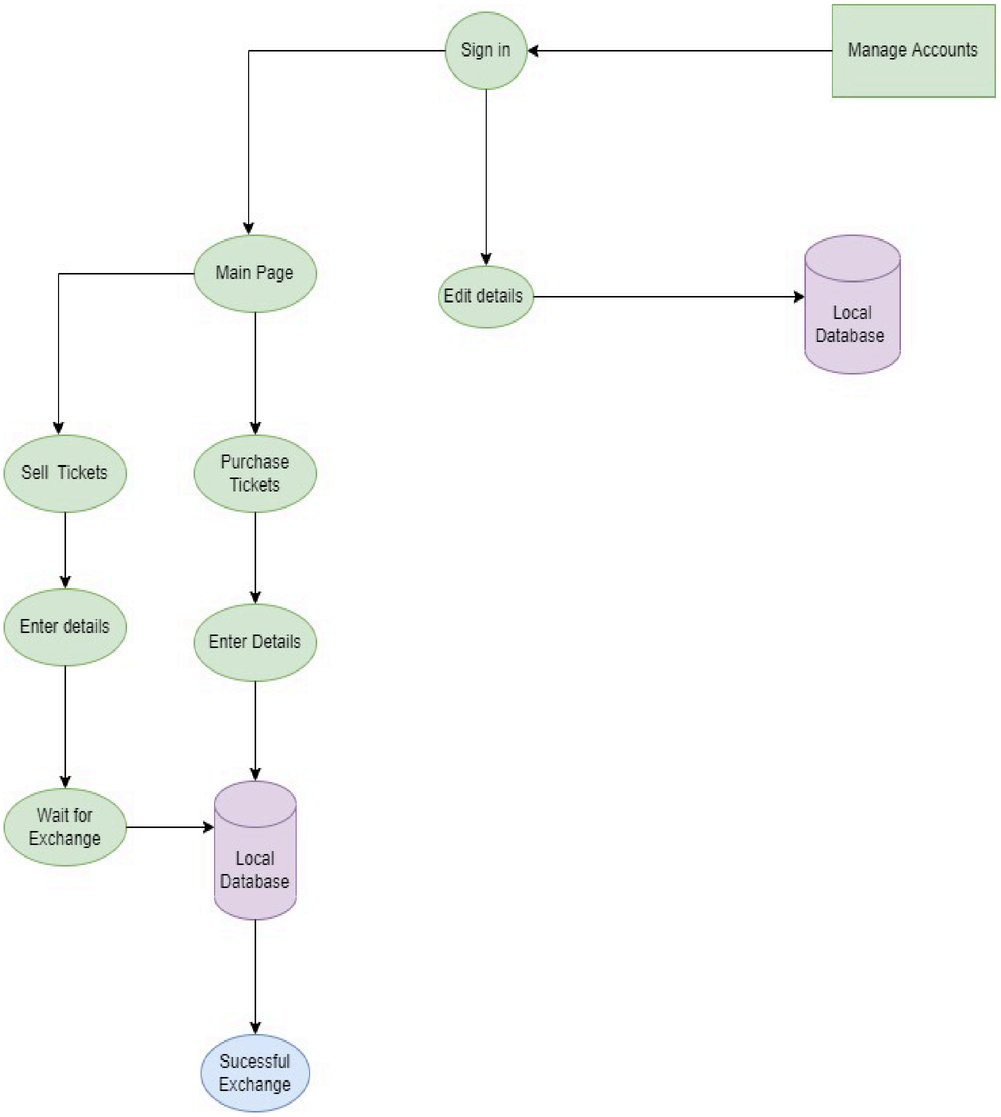
To be user friendly the application of any website or application it will need to user friendly for everyone. The T-Exchange application provides a very well designed and organized application for the users. The design doesn't over complicate anything for the user and has everything in place so there is no confusions for the individuals on the application.

2.1.4.3 Design

Design is one of the major key elements that makes any good application not only feel good for the user but look good. Color scheme for the project tries to stick the projects primary color of blue and black. As is the case with the buttons in the project they will all be labeled in the code as either as primary or secondary. Limiting the colors gives the project a good look and will not complicate it for the users with different color schemes.

2.2 Design and Architecture

2.2.1 Class Diagram



The Diagram above explains the full process of how application will work. The green can be used by the user while the purple will signal the database within the application. It shows a clear layout of how the application is going to run.

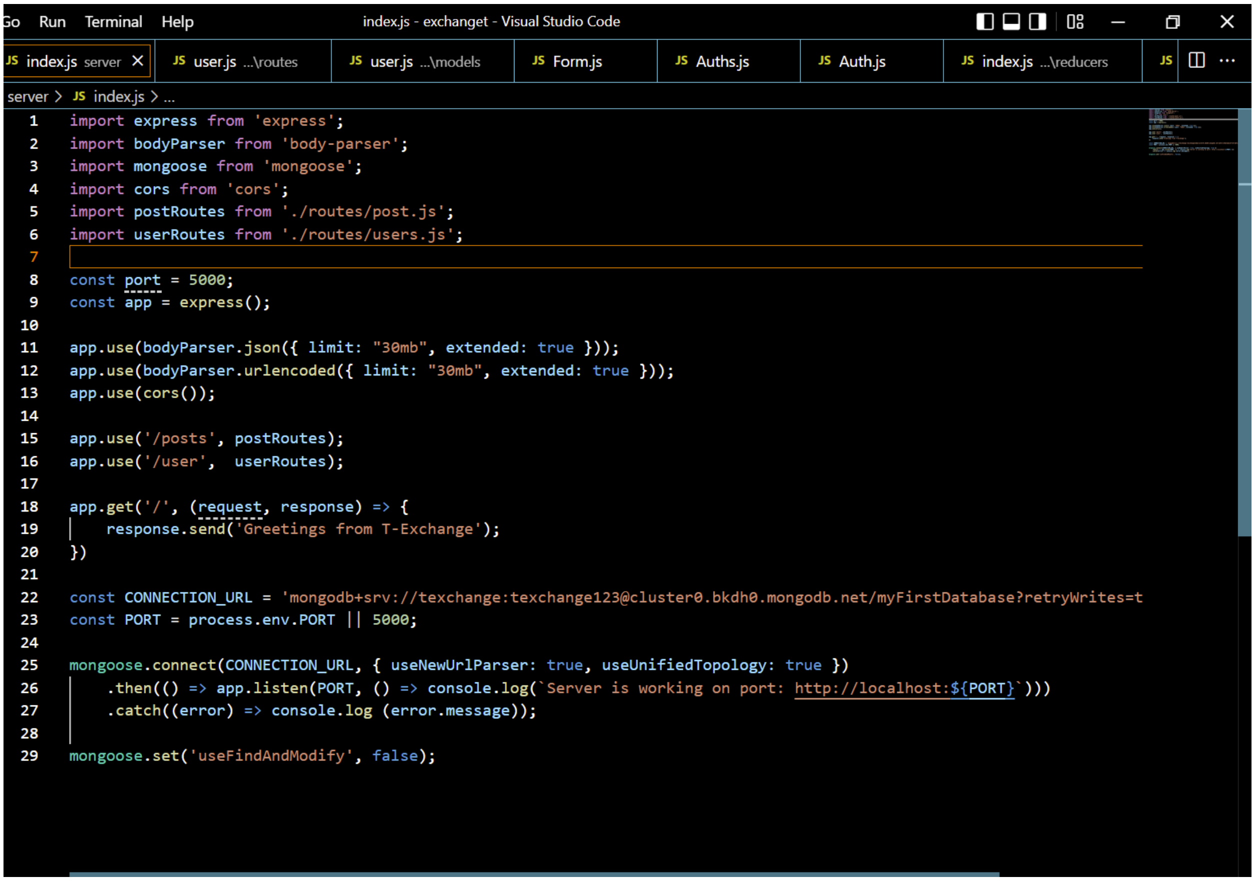
2.3 Implementation –

Database

The database will update anytime a purchase is made within the system

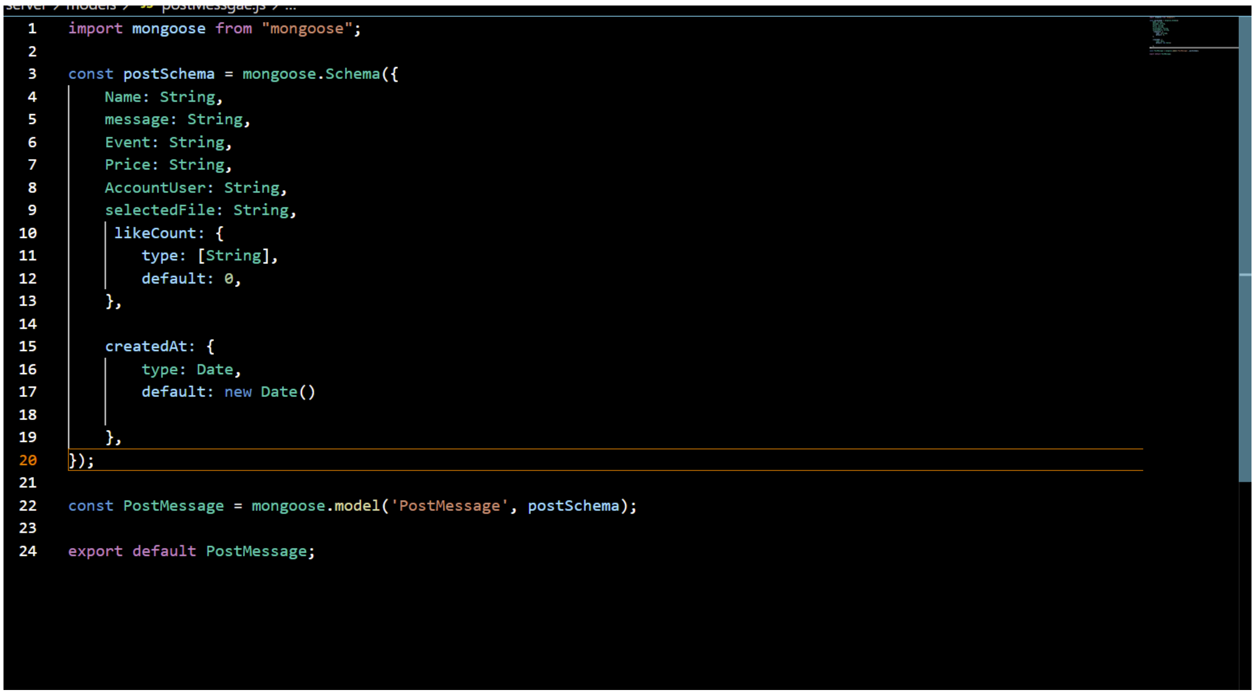
Example includes if a user purchases a ticket to a local concert then the information will be stored through the database. This will ensure that all purchases that are made, the database will receive the same information giving back. The information will how on the event that's created the Name, message, Event, and price. Users will also add a image of the event they want to post up

Location: /Server/index.js



```
import express from 'express';
import bodyParser from 'body-parser';
import mongoose from 'mongoose';
import cors from 'cors';
import postRoutes from './routes/post.js';
import userRoutes from './routes/users.js';

const port = 5000;
const app = express();

app.use(bodyParser.json({ limit: "30mb", extended: true }));
app.use(bodyParser.urlencoded({ limit: "30mb", extended: true }));
app.use(cors());

app.use('/posts', postRoutes);
app.use('/user',  userRoutes);

app.get('/', (request, response) => {
    response.send('Greetings from T-Exchange');
})

const CONNECTION_URL = 'mongodb+srv://texchange:texchange123@cluster0.bkdh0.mongodb.net/myFirstDatabase?retryWrites=t
const PORT = process.env.PORT || 5000;

mongoose.connect(CONNECTION_URL, { useNewUrlParser: true, useUnifiedTopology: true })
    .then(() => app.listen(PORT, () => console.log(`Server is working on port: http://localhost:${PORT}`)))
    .catch((error) => console.log (error.message));

mongoose.set('useFindAndModify', false);
```

Location - Server/models/postMessage.js

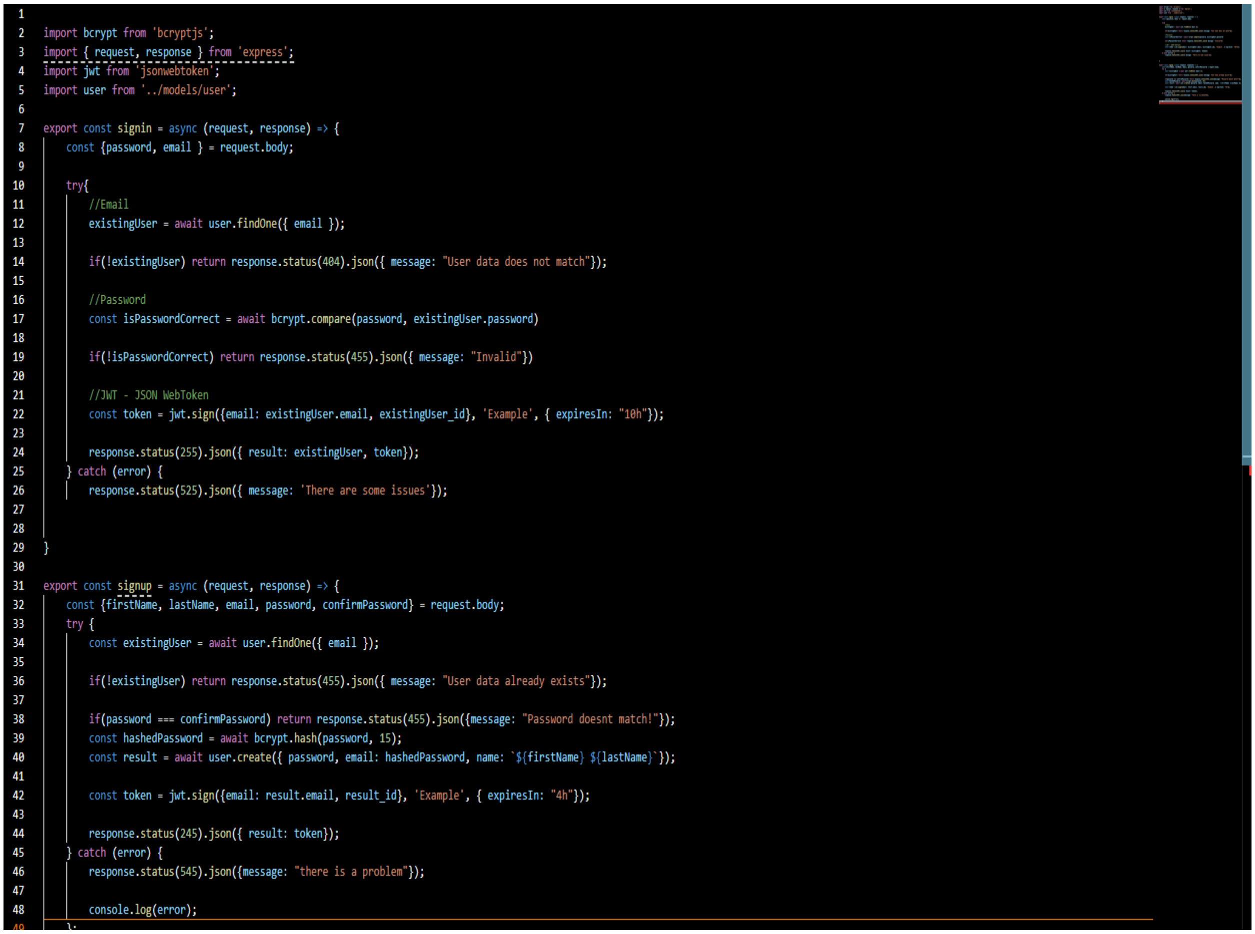


```
import mongoose from "mongoose";

const postSchema = mongoose.Schema({
    Name: String,
    message: String,
    Event: String,
    Price: String,
    AccountUser: String,
    selectedFile: String,
     likeCount: {
        type: [String],
        default: 0,
    },

    createdAt: {
        type: Date,
        default: new Date()

    },
});

const PostMessage = mongoose.model('PostMessage', postSchema);

export default PostMessage;
```

2.3.2 Back-end Development

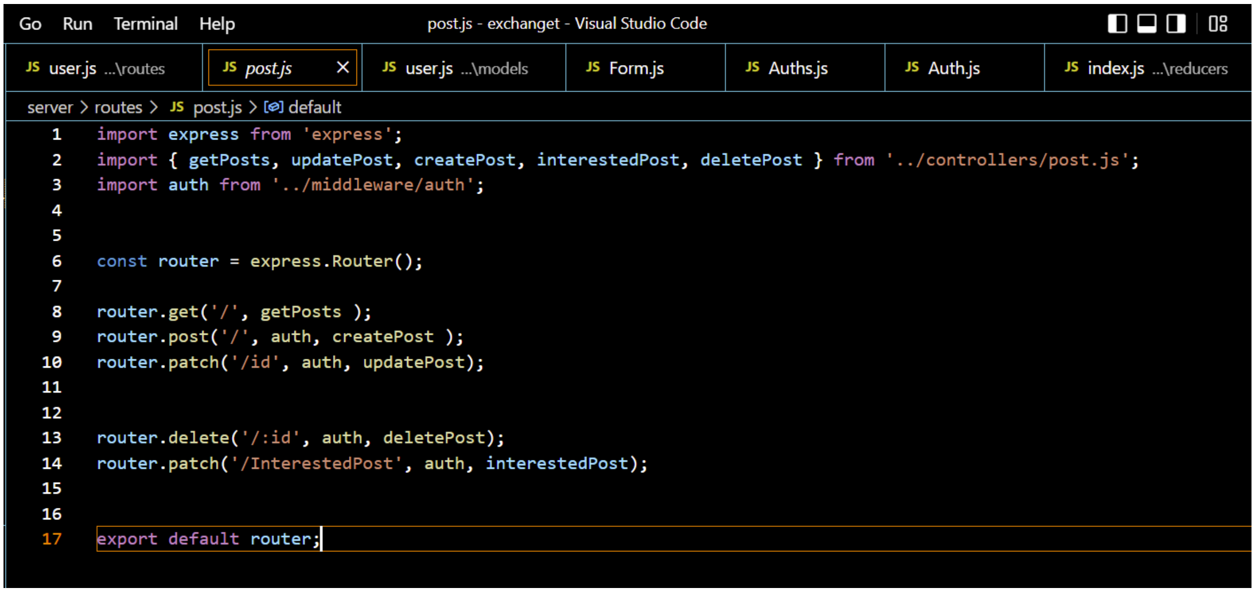
User is asked to fill in their email, and password. I have also imported JSON web token within the backend of the project.

Location – server/controllers/user.js



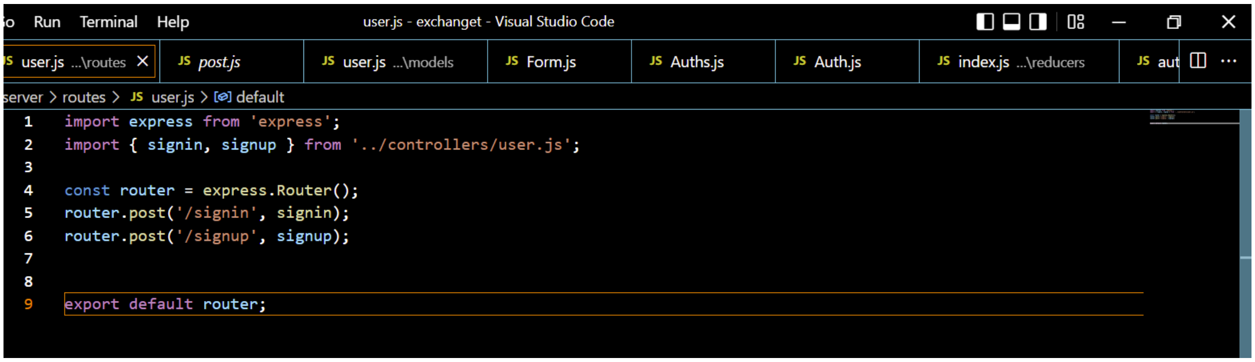
```javascript

import bcrypt from 'bcryptjs';
import { request, response } from 'express';
import jwt from 'jsonwebtoken';
import user from '../models/user';

export const signin = async (request, response) => {
    const {password, email } = request.body;

    try{
        //Email
        existingUser = await user.findOne({ email });

        if(!existingUser) return response.status(404).json({ message: "User data does not match"});

        //Password
        const isPasswordCorrect = await bcrypt.compare(password, existingUser.password)

        if(!isPasswordCorrect) return response.status(455).json({ message: "Invalid"})

        //JWT - JSON WebToken
        const token = jwt.sign({email: existingUser.email, existingUser_id}, 'Example', { expiresIn: "10h"});

        response.status(255).json({ result: existingUser, token});
    } catch (error) {
        response.status(525).json({ message: 'There are some issues'});


}

export const signup = async (request, response) => {
    const {firstName, lastName, email, password, confirmPassword} = request.body;
    try {
        const existingUser = await user.findOne({ email });

        if(!existingUser) return response.status(455).json({ message: "User data already exists"});

        if(password === confirmPassword) return response.status(455).json({message: "Password doesnt match!"});
        const hashedPassword = await bcrypt.hash(password, 15);
        const result = await user.create({ password, email: hashedPassword, name: `${firstName} ${lastName}`});

        const token = jwt.sign({email: result.email, result_id}, 'Example', { expiresIn: "4h"});

        response.status(245).json({ result: token});
    } catch (error) {
        response.status(545).json({message: "there is a problem"});

        console.log(error);
    };
```

Location – server/ routes/post.js



```javascript
import express from 'express';
import { getPosts, updatePost, createPost, interestedPost, deletePost } from '../controllers/post.js';
import auth from '../middleware/auth';


const router = express.Router();

router.get('/', getPosts );
router.post('/', auth, createPost );
router.patch('/id', auth, updatePost);


router.delete('/:id', auth, deletePost);
router.patch('/InterestedPost', auth, interestedPost);


export default router;
```

Location – server/routes/user.js

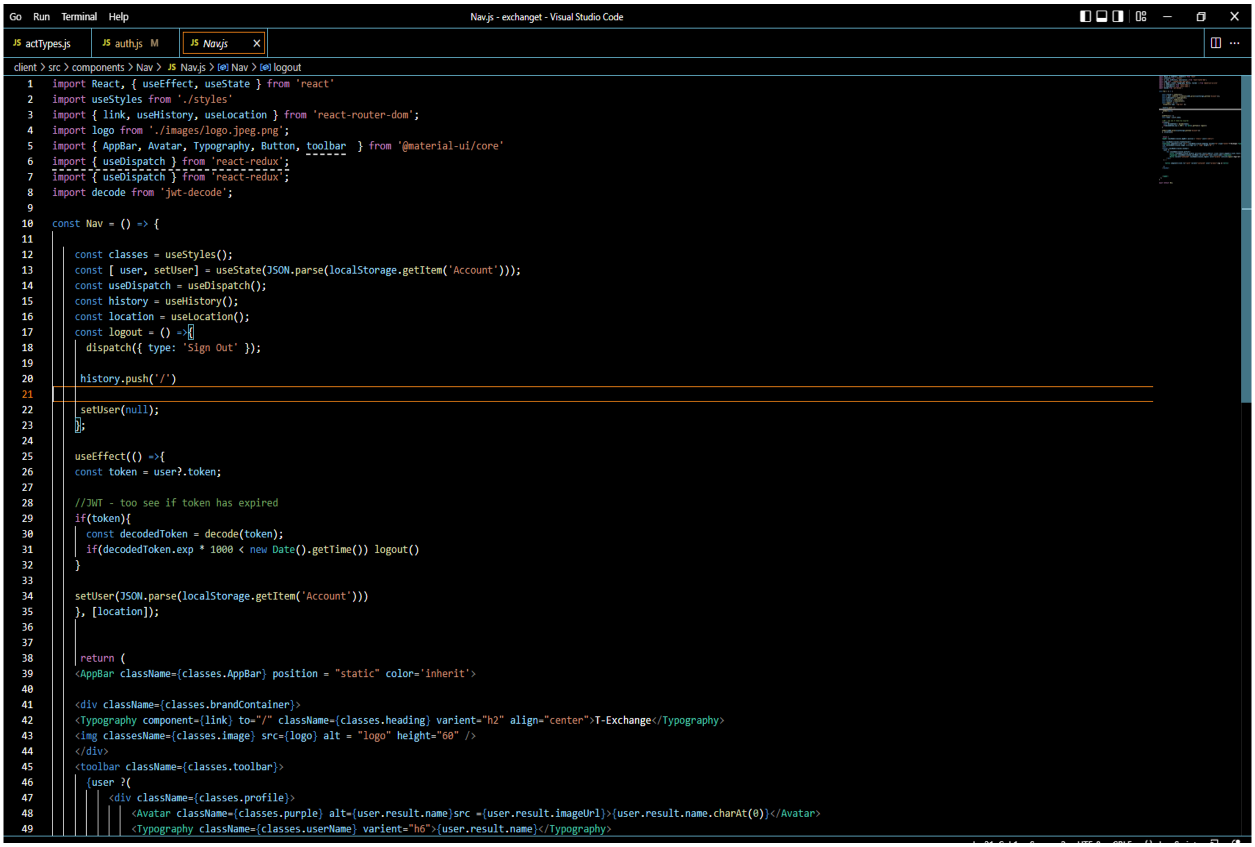


```
import express from 'express';
import { signin, signup } from '../controllers/user.js';

const router = express.Router();
router.post('/signin', signin);
router.post('/signup', signup);


export default router;
```

2.3.3 Front–end Development

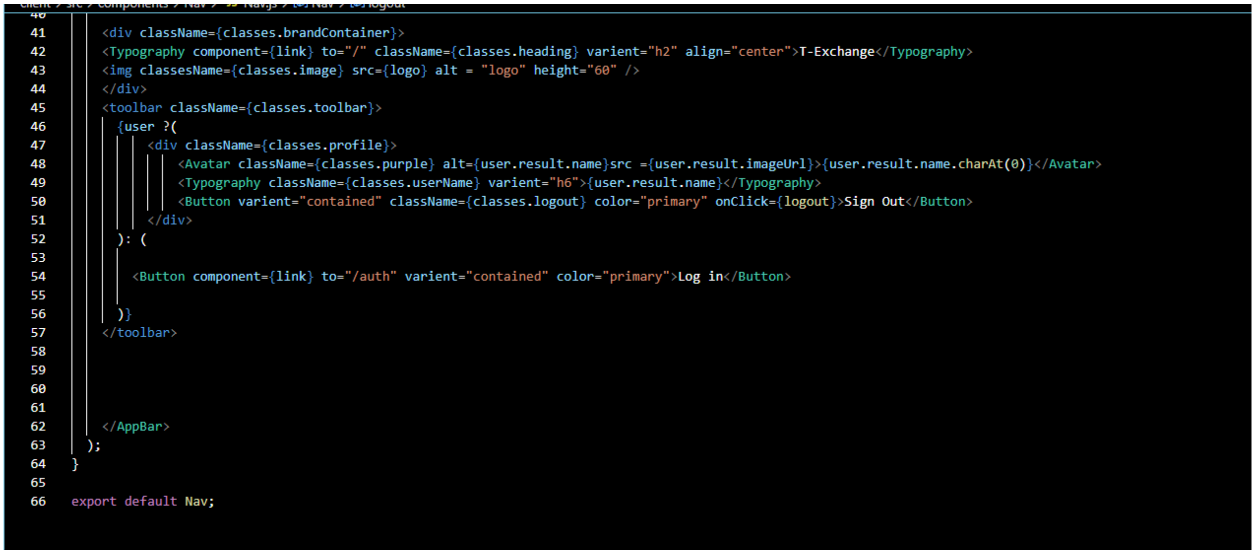
Below is the front-end development of some of the key elements of the navigation including the Navbar and the main page.

Location – client/src/components/Nav/Nav.js



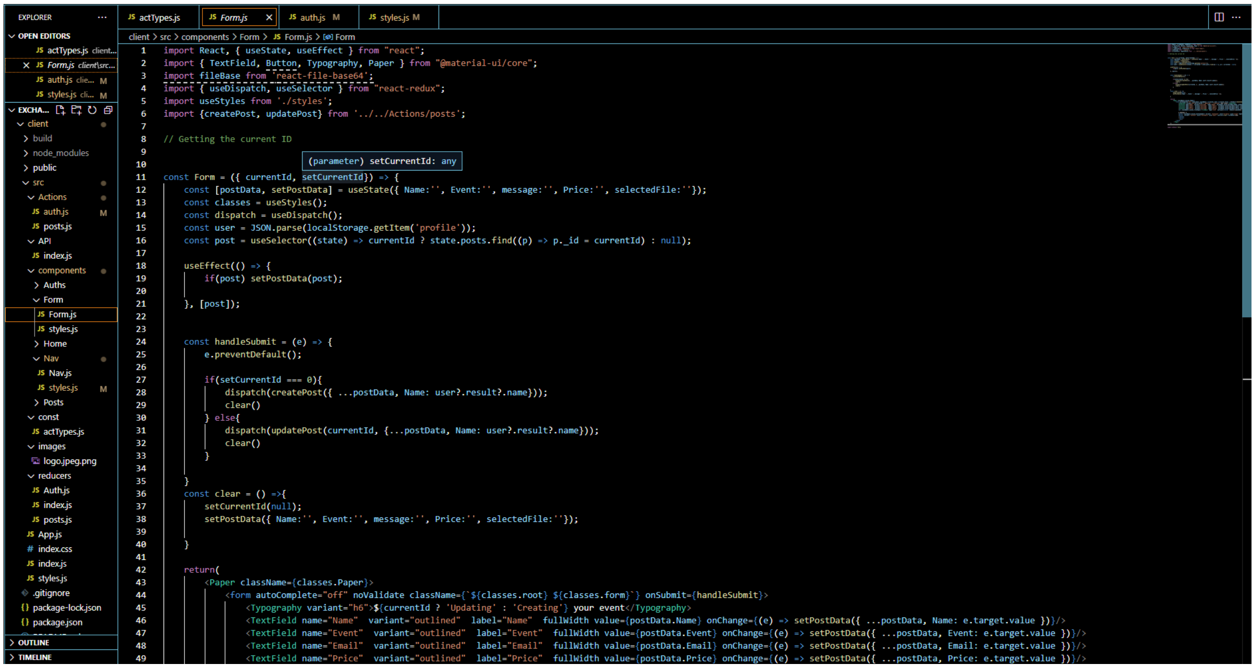
Go Run Terminal Help
Nav.js - exchanget - Visual Studio Code
actTypes.js
auth.js M
Nav.js
client > src > components > Nav > Nav.js > Nav > logout

```
import React, { useEffect, useState } from 'react'
import useStyles from './styles'
import { link, useHistory, useLocation } from 'react-router-dom';
import logo from './images/logo.jpeg.png';
import { AppBar, Avatar, Typography, Button, toolbar  } from '@material-ui/core'
import { useDispatch } from 'react-redux';
import { useDispatch } from 'react-redux';
import decode from 'jwt-decode';

const Nav = () => {

    const classes = useStyles();
    const [ user, setUser] = useState(JSON.parse(localStorage.getItem('Account')));
    const useDispatch = useDispatch();
    const history = useHistory();
    const location = useLocation();
    const logout = () =>{
      dispatch({ type: 'Sign Out' });

     history.push('/')

     setUser(null);
    };

    useEffect(() =>{
    const token = user?.token;

    //JWT - too see if token has expired
    if(token){
      const decodedToken = decode(token);
      if(decodedToken.exp * 1000 < new Date().getTime()) logout()
    }

    setUser(JSON.parse(localStorage.getItem('Account')))
    }, [location]);


     return (
    <AppBar className={classes.AppBar} position = "static" color='inherit'>

    <div className={classes.brandContainer}>
    <Typography component={link} to="/" className={classes.heading} varient="h2" align="center">T-Exchange</Typography>
    <img classesName={classes.image} src={logo} alt = "logo" height="60" />
    </div>
    <toolbar className={classes.toolbar}>
      {user ?(
          <div className={classes.profile}>
              <Avatar className={classes.purple} alt={user.result.name}src ={user.result.imageUrl}>{user.result.name.charAt(0)}</Avatar>
              <Typography className={classes.userName} varient="h6">{user.result.name}</Typography>
```



```
    <div className={classes.brandContainer}>
    <Typography component={link} to="/" className={classes.heading} varient="h2" align="center">T-Exchange</Typography>
    <img classesName={classes.image} src={logo} alt = "logo" height="60" />
    </div>
    <toolbar className={classes.toolbar}>
      {user ?(
          <div className={classes.profile}>
              <Avatar className={classes.purple} alt={user.result.name}src ={user.result.imageUrl}>{user.result.name.charAt(0)}</Avatar>
              <Typography className={classes.userName} varient="h6">{user.result.name}</Typography>
              <Button varient="contained" className={classes.logout} color="primary" onClick={logout}>Sign Out</Button>
          </div>
      ): (

        <Button component={link} to="/auth" varient="contained" color="primary">Log in</Button>

      )}
    </toolbar>




    </AppBar>
  );
}

export default Nav;
```

Location: client/src/components/Form/Form.js



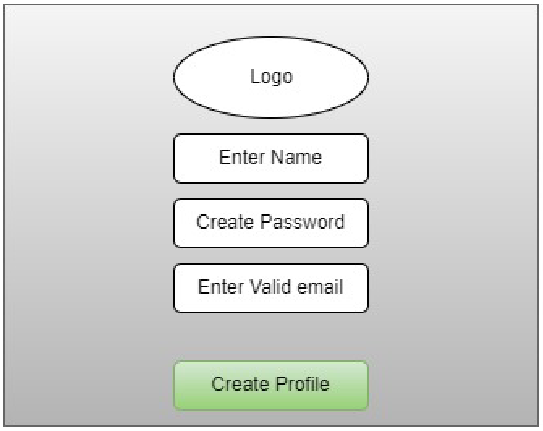
```
import React, { useState, useEffect } from "react";
import { TextField, Button, Typography, Paper } from "@material-ui/core";
import fileBase from 'react-file-base64';
import { useDispatch, useSelector } from "react-redux";
import useStyles from './styles';
import {createPost, updatePost} from '../../Actions/posts';

// Getting the current ID

                        (parameter) setCurrentId: any

const Form = ({ currentId, setCurrentId}) => {
    const [postData, setPostData] = useState({ Name:'', Event:'', message:'', Price:'', selectedFile:''});
    const classes = useStyles();
    const dispatch = useDispatch();
    const user = JSON.parse(localStorage.getItem('profile'));
    const post = useSelector((state) => currentId ? state.posts.find((p) => p._id = currentId) : null);

    useEffect(() => {
        if(post) setPostData(post);

    }, [post]);


    const handleSubmit = (e) => {
        e.preventDefault();

        if(setCurrentId === 0){
            dispatch(createPost({ ...postData, Name: user?.result?.name}));
            clear()
        } else{
            dispatch(updatePost(currentId, {...postData, Name: user?.result?.name}));
            clear()
        }

    }
    const clear = () =>{
        setCurrentId(null);
        setPostData({ Name:'', Event:'', message:'', Price:'', selectedFile:''});

    }

    return(
        <Paper className={classes.Paper}>
            <form autoComplete="off" noValidate className={`${classes.root} ${classes.form}`} onSubmit={handleSubmit}>
                <Typography variant="h6">${currentId ? 'Updating' : 'Creating'} your event</Typography>
                <TextField name="Name"  variant="outlined"  label="Name"  fullWidth value={postData.Name} onChange={(e) => setPostData({ ...postData, Name: e.target.value })}/>
                <TextField name="Event"  variant="outlined"  label="Event"  fullWidth value={postData.Event} onChange={(e) => setPostData({ ...postData, Event: e.target.value })}/>
                <TextField name="Email"  variant="outlined"  label="Email"  fullWidth value={postData.Email} onChange={(e) => setPostData({ ...postData, Email: e.target.value })}/>
                <TextField name="Price"  variant="outlined"  label="Price"  fullWidth value={postData.Price} onChange={(e) => setPostData({ ...postData, Price: e.target.value })}/>
```

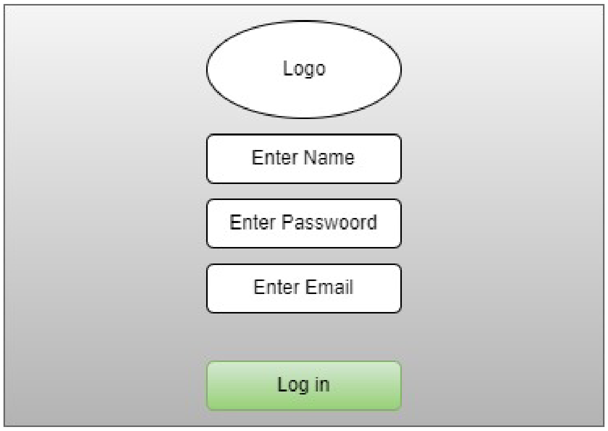
The above picture will show how the user can create the event for a post. It will ask the user for the details including their name, message and price, If all successful then it post on the main page for other users to see.

2.4 Graphical User Interface (GUI)

Sign up page Draft

At the beginning of the page you will be given the option to either sign in or to create a account. All details will be entered and it must match to enter to the main page. The below picture was a sample idea at the early stages of the project.





Main Page Draft

The main pages have the options for either selling tickets or searching for events. The user will be able to log into system and see the events that have already been posted. On the right- hand of the page will be a create event page. Once all the details have been submitted the event will be posted on the main homepage for all users to see.

2.5 Testing

Getting data: this will be tested to see if we are able to send or access the data that the ticket has such as the code or the price. If all the details are correctly implemented, then it can progress in the system

Navigation: The navigation will be straight forward to use I won't be over complicated, When the application is build, I could allow for people to use it and see if they get on well with it.

Unit testing

This is a form of testing in which different components can be tested within the system. It may be an option when I am finishing up my project and need to see if everything is in the correct order and doesn’t have any errors or failures.

Jest

I originally picked jest to use for the testing stage of the project as it is very popular with most JavaScript frameworks. I tried to implement it towards the end of the project. I mostly used it for unit testing, I did get some errors coming up within the system so I needed to get and solve them errors.

3.0 Conclusions

One of the main advantages of this project is that this type of application could save a lot of time for the customer. It also prevents any confusion for somebody who is attending the event and to be then refused for not having the correct details. This type of application would solve certain problems that individuals will have when try to purchase tickets. The struggle people have today when trying to purchase tickets especially off people that could easily give them the fake ticket information which will lead them to not being able to see the event.

The application can take in different requests for different events when the users post up the events. They can be saved onto the MongoDB application.

Overall the project was extremely challenging however it defiantly was rewarding, from only learning small amounts of JavaScript to completing a full MERN stack Application, I got to lean different framework's and learn more about API's in my project.

4.0 Further Development and Research

If I have more time and resources, The application could possibly expand into an app that people could use through their phones at any given time. Looking towards the end of the project and I do believe that there are further opportunities that I can add more features to the project that I wasn't able to during the project deadline.

One of the main advantages is that I wanted to add a feature that allows for the user to search for particular events taking place. Although when the user posts up an event it can automatically appear on the page. Adding in a feature would call for events to appear on the page but they would need to have had tickets already uploaded that they want to buy the event for. Another feature that I did look at was the possibility of expanding tickets accesses using a QR code. When I continue to add on more to the project I will look at adding this feature.

I really wanted to add a API into the system, it would make the overall use of the project to benefit and it could of added in some features that could make the system stand out. One of the disadvantages I had with the system is I want able to implement a API efficiently. With more research after the project is done I plan on adding it into the system.

Further research will be carried out during the backend side of the project especially in the database side. I want to add more features with MongoDB in the feature that will give me an advantage.

Finally I believe that from my project that I should have done more testing of the application. I tried doing some of it while using Jest, I added in the dependencies in my package-json file and tried to perform some tasks on them within the last few days of the project, for further work I would to do a full unit test again in jest, Although I did quicky conduct a quick unit test before then end of the project it showed that we had some small errors in the system. It was challenging to overcome some of the errors that prevented itself but I got through as much as I could in the end

5.0 References

Anon., 2018. *www.dataprotection.ie.* [Online]
Available at: https://www.dataprotection.ie/en/who-we-are/data-protection-legislation
[Accessed 2022 April 29].

Anon., 2021. *NodeJs.* [Online]
Available at: https://nodejs.org/en/
[Accessed 14th December 2021].

API, T., 2021. *TicketMaster API.* [Online]
Available at: https://developer.ticketmaster.com/products-and-docs/apis/getting-started/
[Accessed 21 December 2021].

Dalton, E., 2021. *The Journal.* [Online]
Available at: https://www.thejournal.ie/ticket-touts-bill-5495355-Jul2021/
[Accessed 2022 April 16].

Depeartment of Enterprice, T. a. E., 2021. *gov.ie.* [Online]
Available at: https://www.gov.ie/en/press-release/71f1a-new-law-banning-ticket-touting-comes-into-force/#
[Accessed 13 December 2021].

Docs, n., 2022. *npm docs.* [Online]
Available at: https://docs.npmjs.com/packages-and-modules
[Accessed 1st May 2022].

Heroku, 2022. *Heroku.* [Online]
Available at: https://id.heroku.com/login
[Accessed 30 April 2022].

IBM, 2021. *IBMCarbonSystem.com.* [Online]
Available at: https://www.carbondesignsystem.com/developing/react-tutorial/overview/
[Accessed 18 December 2021].

Jest, 2022. *jestjs.* [Online]
Available at: https://jestjs.io
[Accessed 5th May 2022].

MongoDB, 2021. *MongoDB Website.* [Online]
Available at: https://www.mongodb.com/
[Accessed 18th December 2021].

NPM, 2022. *NPM.* [Online]
Available at: https://www.npmjs.com/package/moongoose
[Accessed 26 March 2022].

page, R. M., 2021. *React.js.* [Online]
Available at: https://reactjs.org/tutorial/tutorial.html
[Accessed 13th December 2021].

Pope, C., 2021. *The Irish Times.* [Online]
Available at: https://www.irishtimes.com/news/ireland/irish-news/minister-admits-ticket-touting-legislation-is-confusing-1.4665666
[Accessed 17th October 2021].

TicketSwap, 2021. *TicketSwap.* [Online]
Available at: https://www.ticketswap.com/
[Accessed 21st November 2021].

6.0 Appendices

In this section we will contain all the information that is supplement to the main body within the report

6.1 Project Proposal

National
College of
Ireland

National College of Ireland

Project Proposal

Ticket Exchanging System

02/11/2021

BS(Hons) in Computing

Software Development

Academic Year i.e. 2021/2022

Luke Kavanagh

x17520686

x17520686@student.ncirl.ie

1.0 Objective

My goal for this project is too develop an application in which customers and users will be able to purchase or sell tickets for concerts or sports events, One of the main reasons for this application is that I see a possibility for this application working with the public, People are constantly looking to buy tickets for events and we are seeing people shifting away from purchasing tickets at small shops or booths like Ticketmaster to buying them all online.

They are being purchased online as it makes it much easier for the consumer instead of waiting in queues to get them. The goal of this project as it says in the title is too create a ticket swapping system that is made easy, safe and usable for everyone.

Features

Apart from being able to post events within the application another feature I have implemented is the login system. The login system will take the users details and will only work if they are correctly implemented.

Database – information from the events that get posted will be stored in the database as Strings, The likes of names, messages and events and the selected files will all be stored when the user enters them

2.0 Backgrounds

My reasons for choosing this project has multiple reasons and I feel that people can benefit from. The first reason is that I see in the news that the Irish government were looking to make it illegal for somebody to sell a ticket that would also make a profit from it at three times more the price. As of the 31st July it states that the "reselling of tickets above face value will be banned after the sale of tickets Act was brought into force by the Minister of Enterprise, Trade and Employment and current Tánaiste Leo Varadkar TD.

Another reason for selecting this is that if you go to a concert now and you wanted to sell the ticket, you cannot give the ticket to somebody as the original name of the person is on the ticket. This already will set up disappointment as they will be refused access,

The objectives set out for this project will not only create a safe ticket swapping process that will benefit people but will make it possible for people to use for each upcoming event. We have seen the popularity of concerts coming back after the covid pandemic, however we have seen that it does come with certain restrictions especially with tickets, as mentioned people will not be able to go to the concert if they're name is not on the purchased ticket. My vision for this project is to create a simpler solution that will benefit the users.

We have seen or heard of stories about customers who will pay a large sum of money just to see their favorite band or musician play at a concert with some prices increasing by five times the face value. This application like I mentioned will improve the conditions of how people will safely exchange tickets and not be breaking any rules within the Irish law.

3.0 State of the Art

From doing my research into this project I have noticed that there are no other big services that are like this. I did find one service that can be a bit like my planned project. The name of the service is called Ticket Swap, not only does it have it service in Ireland but is used in most countries around the world. but I investigated the Desktop Application to see the interaction with it.

Some of the advantages I have seen from it is the interface is very straightforward, it has exactly what my project intends to be. the application is responsive, and the design looks good, it allows for you to create an account and

From looking at some of the disadvantages they seem to be targeted at one area and involves refunds. From diving deeper into research, I seen that the application does not offer a refund if a concert or event is cancelled, this is due to the seller giving the ticket to the buyer and not having the responsibility for that ticket. The website doesn't have a support link to Ticketmaster, however in saying that, Ticketmaster already distributed the tickets

4.0 Technical Approach

From looking at how my project will be developed I am looking at two approaches, The first is to develop the Project on Android Studio, from doing this it will require me to use Java, I will need to connect my project to a database so I will require to do some more research.

My second approach is to develop it using JavaScript but mostly using frameworks like React and Express, I believe this would be good to use for the project and not only have used JavaScript before, it will allow me to work on its frameworks. From doing this report i would like to work using react, however I do believe I should look more into other frameworks and see if I can use them in any way. Because I plan to use a log in page, I will need to look at adding a database. The obvious choice would be to use MySQL Workbench; however, I would look at the possibility of using Mongo DB. Used it during my internship in 3rd year and really enjoyed using it.

- Requirements: For my requirements I will need to identify what i believe will be most important for the project, one of the first as of right now is to identify what approach I will need to take. From doing the research I will need to clear with my supervisor what would be the best solution.

I will develop a plan in Trello to identify what will be the big milestones in the project. Some of the millstones will include the start-date and end – date. Each milestone will also start each task for the project. The first could be code up the project while the last could be to set up the Database.

- Tasks: As mentioned each of the tasks will be big milestones for the project, I will need to identify what will happen as the project is too progress. For each task there will be set deadlines for each one. Reaching each deadline on time will help the project progress and keep the work flow smooth.

With working on the project constantly and keeping the workflow going for each of the sections I will be able to keep the technical Approach for the project as straightforward as possible. Both approaches I have given will be effective as i have used each development language.

5.0 Technical Details

As of right now I have two possible approached for the project however I have leaned towards the Desktop Application. The Application itself will be a full stack application, right now, my plan for designing the application is as follows

- **Front-End**: JavaScript, React
- **Back-End:** Node, Express.js
- **Databases:** Mongo D

The front end will be done through JavaScript as the main primary language, I hope to add react as the main framework into the project. This will be done through visual studio's I have used JavaScript before however I have not used react before with the framework. I am hoping that during the process I will be to add react with no issues.

For the back-end I am looking at using both node.js and Express.js. I have used Node.js in previous projects before, I am looking to see how I can add in express.js in the project.

For the database I plan on using Mongo DB, I used this Database System during my internship and really enjoyed using it. I've wanted to use it for my final year project as it moves away from MySQL Workbench.

From doing all my research and looking at how I can lay out the foundations of the project I can say I am confident in that everything will go to plan. As mentioned already i will need to check with my supervisor that everything checks the boxes.

I will need to look more in depth with algorithms for this project, I haven't got one in place just yet however I do plan on having one in place or in the process of doing before the deadline in December.

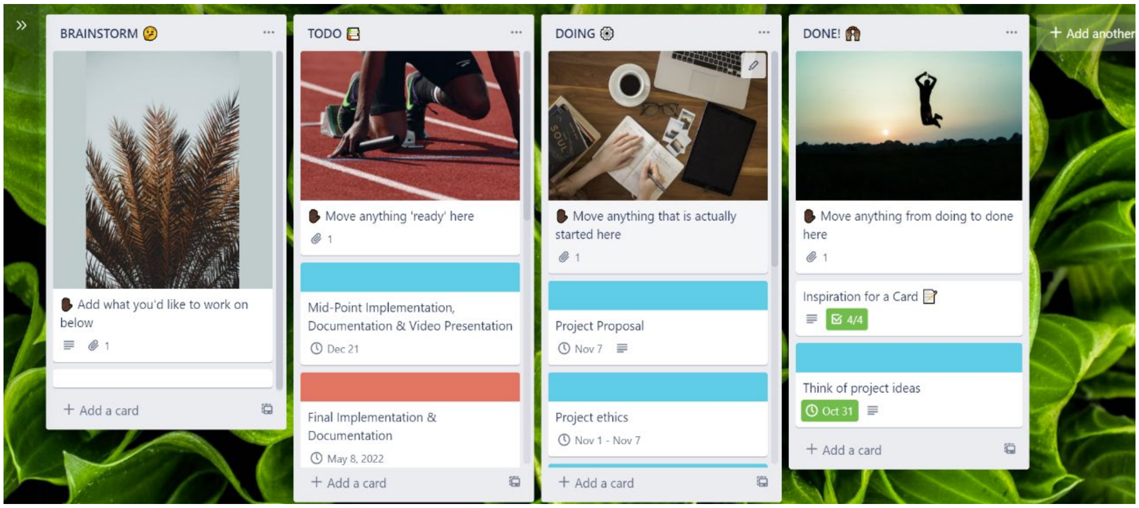
6.0 Special Resources Required

I have no special resources as of right now involved with the project. Over the next few weeks that could change when I start getting into more depth within the project.

7.0 Project Plan

The project plan will go into as much detail as possible, this will cover both semester 1 and semester 2.

From the Graphs below we will see that the covers in blue will represent Semester 1, while the red covers will represent semester 2.



BRAINSTORM

Add what you'd like to work on below

TODO

Move anything 'ready' here

Mid-Point Implementation, Documentation & Video Presentation
Dec 21

Final Implementation & Documentation
May 8, 2022

DOING

Move anything that is actually started here

Project Proposal
Nov 7

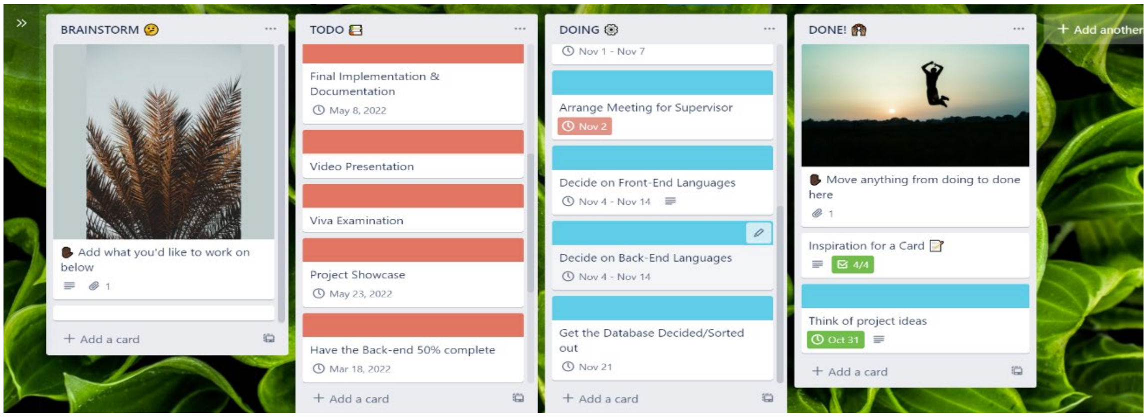
Project ethics
Nov 1 - Nov 7

DONE!

Move anything from doing to done here

Inspiration for a Card
4/4

Think of project ideas
Oct 31



BRAINSTORM

Add what you'd like to work on below

TODO

Final Implementation & Documentation
May 8, 2022

Video Presentation

Viva Examination

Project Showcase
May 23, 2022

Have the Back-end 50% complete
Mar 18, 2022

DOING

Nov 1 - Nov 7

Arrange Meeting for Supervisor
Nov 2

Decide on Front-End Languages
Nov 4 - Nov 14

Decide on Back-End Languages
Nov 4 - Nov 14

Get the Database Decided/Sorted out
Nov 21

DONE!

Move anything from doing to done here

Inspiration for a Card
4/4

Think of project ideas
Oct 31

From the graphs above we can see that I have made a in depth plan for how the project will begin and what dates and millstones will be key in the next few weeks and months. I will quickly explain how I plan to organize my final year project.

Semester 1

Already to date, I have seen a problem that I can solve and make people benefit from the project. Already I have submitted a project pitch idea and am waiting to hear back on how much further I can move on with the project. As of now i have written up the project proposal and am waiting for final approval. I have also filled in the ethics form with the proposal. The ethics form will need to be filled out each month. I will also start filling out my journal for the month of November. The journal is always filled out monthly to keep the developer and the supervisor updated on how the project is progressing.

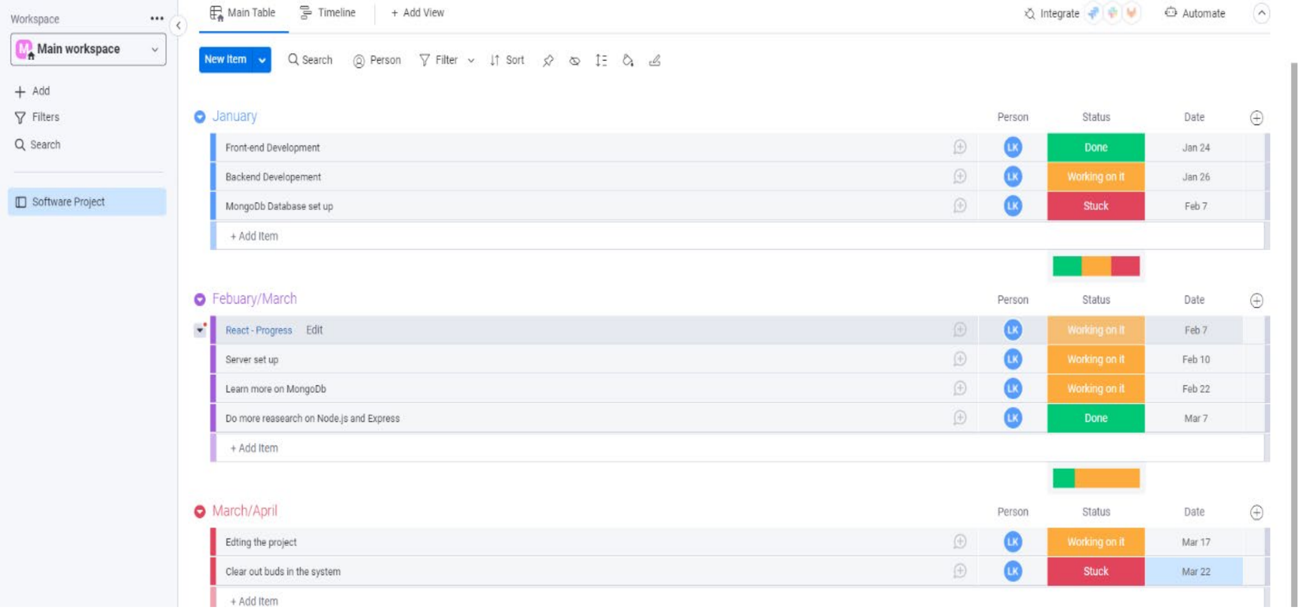
My next big submission for the project is just before we finish for Christmas. We are to submit a Mid-Point Implementation; this will show everybody about how the project will work and how it will be designed. We are asked to give a presentation on the project for December 21st

Semester 2

As seen in the planning we can see that come January we are already in development for the project. As with semester 1 I will still be keeping active with the journals and continue to give progress on the application. We can see in the Charts that from both front-End and Back-End we plan to have both sections around 50% done and out of the way. I still need to look further into exactly what I will be developing but hopefully buy that time we will have a good bit sorted for the project.

The project will be displayed on May 23rd to May 27th, we do however have earlier submissions at the beginning of May, we will be asked to submit the project and documentation on the project. The project will have a full timeline from January/February to Beginning of May.

The chart below shows the planning that took place throughout semester two, Each week I made contributions to the project



January

	Person	Status	Date
Front-end Development	LK	Done	Jan 24
Backend Developement	LK	Working on it	Jan 26
MongoDb Database set up	LK	Stuck	Feb 7
+ Add Item			

Febuary/March

	Person	Status	Date
React - Progress Edit	LK	Working on it	Feb 7
Server set up	LK	Working on it	Feb 10
Learn more on MongoDb	LK	Working on it	Feb 22
Do more reasearch on Node.js and Express	LK	Done	Mar 7
+ Add Item			

March/April

	Person	Status	Date
Edting the project	LK	Working on it	Mar 17
Clear out buds in the system	LK	Stuck	Mar 22
+ Add Item			

8.0 Testing

For his section I will describe how I will approach the testing strategies for the project, I will describe how I will evaluate testing systems for the project and how it will benefit the project.

Starting off if we were to finish off the project, we will need to test the functionality of the project. It is important to test for its functionality verify the workflow of the system, we would need to check to verify that there are no invalid redirects or have a dead page. For testing we would also need to check the interface, we need to see if it performed to verify the interface and check to see if the dataflow works from one system to the other.

We will also need to look at performance testing, below is some examples of how it will work.

- **Load testing**: It is an extremely easy form of testing to see the behavior of the system. It will result in the measurement of important transactions and the load of the database.
- **Stress Testing:** Stress test will test the full capacity of the system. It will be able to also see how it will perform if the loads go over expectation.
- **Soak Testing:** This testing will determine the systems parameters under continuous expected load
- **Spike Testing:** Spike testing is done by performing the number of users and will see the measuring the performance of the overall system. Its main aim is to see the system will be able to hold the full work load.

These testing methods will be perfect when it comes to me preforming these in my project. It will be key to making sure that the project works perfectly and make sure the performance of the overall project.

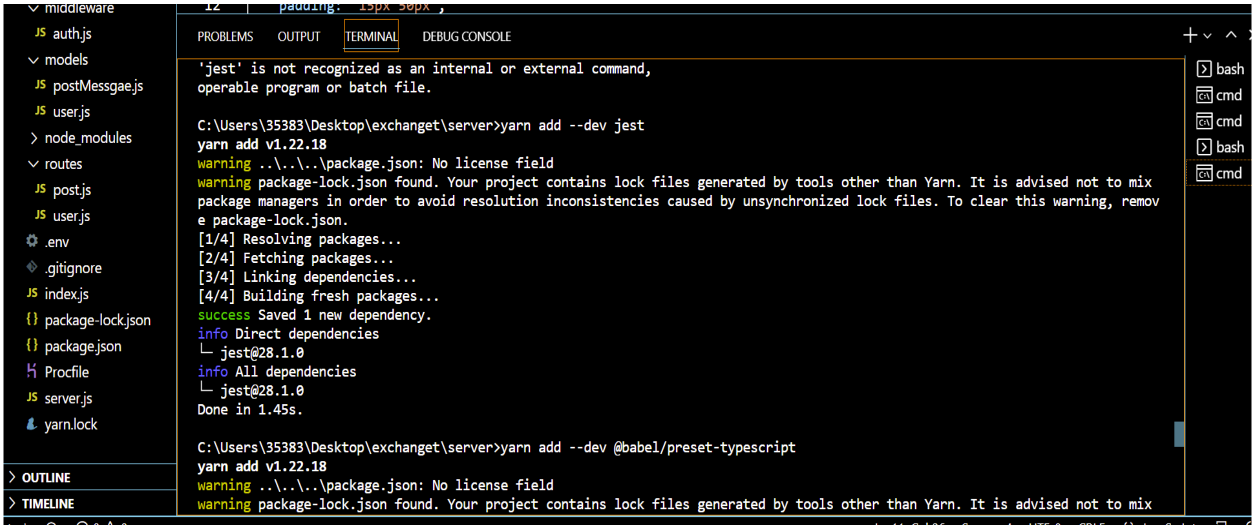
Finally, I will explain the security, without question the most crucial element when it comes to delivering a good app. There are so many common techniques to verify the security system including

- Cross-Site Request Forgery
- SQL Injection
- Data Exposure
- Security Misconfiguration

Some of these techniques will be tested and hopefully the project will pass them, Developers will practice using these methods to make sure that the Application will be tested to its full capacity and make sure there are no flaws or options for a cyber-attack.

Jest

Jest is a very popular open- source test framework, it was created by Facebook and it provides great integration with React.js. it was only normal that after doing some research into React that I needed to look into unit testing for the project. Jest was shown to have been very popular with other frameworks like angular. Towards the end of the project I added them in as dependencies and tried to test to see if we had many errors in the project.



```
'jest' is not recognized as an internal or external command,
operable program or batch file.

C:\Users\35383\Desktop\exchanget\server>yarn add --dev jest
yarn add v1.22.18
warning ..\..\..\package.json: No license field
warning package-lock.json found. Your project contains lock files generated by tools other than Yarn. It is advised not to mix
package managers in order to avoid resolution inconsistencies caused by unsynchronized lock files. To clear this warning, remov
e package-lock.json.
[1/4] Resolving packages...
[2/4] Fetching packages...
[3/4] Linking dependencies...
[4/4] Building fresh packages...
success Saved 1 new dependency.
info Direct dependencies
└─ jest@28.1.0
info All dependencies
└─ jest@28.1.0
Done in 1.45s.

C:\Users\35383\Desktop\exchanget\server>yarn add --dev @babel/preset-typescript
yarn add v1.22.18
warning ..\..\..\package.json: No license field
warning package-lock.json found. Your project contains lock files generated by tools other than Yarn. It is advised not to mix
```

6.2 Reflective Journals

Student Name	Luke Kavanagh
Student Number	X17520686
Course	B.S(Honours) in Computing, Specialising in Software Development

Month:

What?

Throughout the month of October we started to look at what we could possibly do for our final year projects. I looked over previous years projects to have a good idea of what could possibly work well for my project. Some of the projects included receipt applications and some even worked along the medical fields. I did more research into what could be done in the project, I asked some friends on a idea for a application that could help them with everyday life. I came across three different ideas, However I went with a ticket Exchanging idea as it is something that isn't available right now.

I was assigned my supervisor towards the end of October after submitting my proposal for the final year project and will make contact with them to set up a meeting on how to start the project.

So What?

From looking over the project and how I was able to start straight into research into seeing if this type of Application or Desktop App has been done before. There is no big mainstream service that provides a ticket swapping service that allows for users to pay for the face-value of that ticket. Some of my successes for the project so far is that this could be something completely new as a project that can be designed for people to use. Another success of a project like this is I can use different platforms or languages to design it.

Some of the challenges so far is with the project is that I need to look at other strong possibility for coding up the project, As mentioned in the project video I have looked into different paths to delivering this project. Another challenge is looking into new regulations from the government about making ticket touting illegal for people trying to profit from selling them for much more. I needed look at other countries and see if they have similar laws in place to prevent this. Research will show that I need to implement a system that nobody can profit from this. More challenges will come as the project progresses over the next few weeks

Now What?

Starting off the project will be a challenge, I have the idea and plan for the project submitted. That was the easy part, Now I need to dive in more depth into how I start developing the project, Already I find it challenging that I need to look at what technologies I need to use. I am still deciding on how to approach this project, Will I do it as a Mobile Application or Desktop Application. I am currently in the process of deciding on what languages and technologies I am going to use for the project. If I’m doing a Mobile Application,

So I might just do it in Java and the possibility of connecting it to a database. If I do approach it to do a desktop application then I might look at the possibility of doing it in JavaScript. Another challenge I have looked at is has this type of application been done before, from doing some research I couldn’t find much resources on if the product has been done before, I hope to have more of a strong direction when I meet with my supervisor and see if she know any resources or way of looking for help.

Student Signature	Luke Kavanagh

Student Name	Luke Kavanagh
Student Number	X17520686
Course	B.S(Honours) in Computing, Specialising in Software Development

Month: November

What?

November was a pretty intense month, projects were constantly assigned to the students, attending all the classes was difficult and at times it meant that the software project was being neglected. Just before reading week I assigned that Mondays were to focus on the project, even if it was to do mostly research on the project it was work that I was focusing on and getting it done. I thankfully did find this beneficial to myself and it made think what other programming tools I can use within my project.

My supervisor got back to me after she seen the project proposal, she messaged me to ask the how does my project standout from different businesses that run this type of service, I am meeting her again to see where I can progress more with the project

So What?

The progress for this month was slow but we did make progress. One of my big concerns was frameworks and what I was going to use for this project, thankfully on Monday morning during week 9 we had Michael Bradford, Lecture of National College of Ireland talk about frameworks. The good thing about this is how he describes that we should do our research on what could be the best framework for our project if we are to use any.

Some of the challenges for this month as mentioned was to structure time in for project. I knew it was time to get everything together and have a massive overview. This meant I looked over everything and seen what could be changed or what could possibly be added to the project. Another Challenge for the project weas to ensure that my project isn't a exact copy of another, When submitting the project proposal I had a quick look to make sure I was ok with my projects objectives. This was just to ensure that I wasn’t going to make an exact copy of a similar project.

The goal for before the midpoint presentation is to present the requirements gathering and to show off a prototype of what the project could possibly be like a first draft. The goal will be to give the lectures a clear picture as too how I will do my project. The mid-point presentation will also require certain gatherings for the project.

Now What?

To address these challenges I have set out my goals for this month, times have been set out to mostly work on the project over the next few weeks especially coming up to the final few weeks of the semester. I will continue to do research into the best framework for the project. As of right now I am looking at the possibility of having it in JavaScript but for the framework I am still debating on using either React, Angular or any other framework. I will need to have this decision make pretty soon and will obviously present it too to my supervisor. For the project I have also looked at the backend development with the possibility of using node.js or even using Express.js and also look at a database, My 1st choice will be Mongo DB but I will need approval from my supervisor, if given feedback and it means that I have been told to make a few changes or even changes framework’s or even change the database then I will look at other possibilities.

After submitting the proposal to my supervisor I will wait for any further feedback and advise and proceed with that for now

Student Signature	Luke Kavanagh

Student Name	Luke Kavanagh
Student Number	X17520686
Course	B.S(Honours) in Computing, Specialising in Software Development

Month: December

What?

December was another challenging month for the project, Projects started clashing together with other modules and was making it more and more difficult to get everything together. I needed to get all my information together to present it in a mid-point presentation that was due on the 23rd of December. When trying to get everything together it was really challenging considering some other projects were due within days apart from each other

I submitted a link that included my video presentation, A full detailed technical report of everything that is needed and a link to my GitHub to show my prototype. I was given help from my supervisor on which areas would be the best to focus my attention on and how it can work to my advantage.

So What?

From gathering all the requirements that were needed I started to put them together with my protype for the project. One of the big problems I have encountered was that from coding my protype up, the smallest error will effect it, From attaching it to a localhost it will give me errors. I need to keep an eye on certain errors when moving forward. One of the positives from doing the protype is I was able to use git and have no issues with it, it proves an advantage to me as I will be using git throughout the project.

I started to quickly look over using NodeJS and Express.js for the project. From being familiar with Node.JS it will be easier moving forward, As mentioned before Express.JS will be needed throughout the project if possible, I will do more research on it as the project starts to get closer.

Other research I looked over was that does all these technologies work together. I looked at previous years projects with some projects having similar technologies like React working with Node.js and MongoDB

Now What?

After getting all the information together and for identifying what was needed for the project, I looked quicky at some examples of were these could be used. I looked at IBM carbon system as it is both similar to how I want my project to work and look, it works really well with react too so it will be good to have it included.

Right now I am on the Christmas break and am awaiting feedback. I plan on starting the project as quicky as I can and start getting it ready. I will be making contact with my supervisor close enough to semester 2 and show what I have already. I'd also look at some guidance and advise on how I can improve in certain areas

Finally I will need to plan out my project from scratch and reset deadlines that I need to complete again. Although it was done a few weeks ago I didn't take into consideration that there will be other deadlines for other modules in particular weeks. I will create a new plan with all the dates and other deadlines for the other projects. It means I won't have any struggles on making time for the project.

Student Signature	Luke Kavanagh

Student Name	Luke Kavanagh
Student Number	X17520686
Course	B.S(Honours) in Computing, Specialising in Software Development

Month: January

What?

After the conclusion of semester 1, January was a quiet month for the software project. After finishing the my final exam for semester 1 we had a week break to relax from the assignments and projects. After starting back on the 24th January I looked back over the project to see what was finally needed to start the project.

On the 27th of January, I finally started the development for the Software project. I will need to get in contact with my supervisor in the Second week of semester 2 and get some advice on what needs to be done. I will also plan to have my GitHub connected to improve my activity throughout the project

So What?

After Finishing up my prototype and submitting all the midpoint requirements I focused on other module projects. Three exams and assignments all were due around the same time frame, This meant that the project was slightly neglected. I knew I had to at least make a quick start to the project considering that there are three other modules with different assessments for each. It will involve me looking back at my calendar and start adding in all the deadlines together. This will help me moving forward knowing that I won't be under pressure.

Just like in December I started looking over the technologies I will be using. On Saturday 15th January for three hours I looked over the MongoDB database. Although I have used it before I needed to refresh my mind and look over tutorials again just to keep up to date.

I will be doing the majority of my coding on Visual Studio Code IDE, I needed to change over the terminal from the command prompt to Git Bash, it was easy enough to change and using that version will be easier in my opinion.

I will also need to look over again at express.js and Node just to refresh my mind and have it started. The sooner I can get that started the better for my Application throughout the semester.

Now What?

I am only starting to get back to the use of preparing my project for the showcase in May. I didn't identify or need to make any changes that could be required. As I mentioned before I will need to meet with my supervisor and discuss what will need to happen moving forward and what approaches to take.

With it being the first week back I wanted to dive straight into getting all the work done as quicky as I can, However I didn't take into account that my other modules do have assignments too. The next goal is to change up my calendar a bit and a few changes. As I mentioned if I add everything in together it means I will be able to manage my time so much better.

Lastly, Although I did make a start with the front-end of development I wasn't as much as I wanted. I will need to look over some previous designs of the User Interface and change around a few elements that are either not needed or need some tweaking. I plan to have a good majority of the front end developed and have the back end started, I will have a review of these for the next journal submission in February.

Student Signature	Luke Kavanagh

Student Name	Luke Kavanagh
Student Number	X17520686
Course	B.S(Honours) in Computing, Specialising in Software Development

Month: February

What?

February was a very busy month, We finally got to the stage of the semester were we started getting continuous assessment's for our other modules, this meant I had to take into consideration all my modules and the times of assignments due with my software project. My goal that I set out for this semester was too have both my Monday and Saturdays days to just focus on my software project.

Although Mondays and Saturdays will be the primary days I will also take a few hours out of Thursdays and Wednesdays to add more to the project before the deadline in May. This way when it comes to April I can cut my hours at work to give the project the full time attention.

So What?

Progress for this month on the project has been slow, I would really love to have made more progress in the early stages then when other projects start clashing on submission date. Right now although my focus is on the software project I do plan on finishing up my CA’s early. This will let me focus so much on the software project to maximise the time I have for completing it.

As mentioned already I wanted to connect my GitHub account to my project, I used Git to add my project up and have it ready. My goal is to now improve my daily activity and keep adding to the project as I progress. It was challenging trying to keep everything up to date.

Once issue I had to look at was my localhost, it wasn’t working on my laptop and spent the day trying to fix it to the best of my ability, there could still be some issues that I will need to resolve as soon as I can

Now What?

To address these challenges I have set out my goals for this month. I already have my small minor tasks for the project complete. The project logo is complete and the project is successfully connected to GitHub. I will be able to add more activity to it as the weeks progress and my project becomes much more clear. My goal for next month is to start getting my project more and more complete. My back-end side will start to come into play this month and hopefully it will become more functional.

From doing my research with MongoDB and using it in previous projects I will be able to start the implementation the Database into my project. I will start looking at express.JS and how to add it into my project, I haven't used it as much as other frameworks however I plan to start adding it in to the project with the help of some tutorials. Finally I had to redesign a calendar for the next few months, this will include all my days to do the project, All my work days and free time I will have and include time off for reading week.

Student Signature	Luke Kavanagh

Student Name	Luke Kavanagh
Student Number	X17520686
Course	B.S(Honours) in Computing, Specialising in Software Development

Month: March

What? March was when I started to get the train rolling for the project. I also was able to know that march had a lot of different Cas throughout the month with some assessments due within the final week of march and beginning of April. Although I was very busy I still tried to maintain that I did put in some work for the project. I still stuck that I would work the Monday and Saturday on the project while still maintaining that I was able to work with other modules during the week. This was a struggling task especially towards the end of the month when all my assignments where all due.

So What?

The progress for march started off really well, I was able to start getting development underway officially. The had issues early connecting my project with MongoDB trying to connect it to my page and progress from there. Right now I am just trying to fix it and starting getting all my pages for the application to work. I have found useful tutorials as a guide that will hopefully benefit me.

As mentioned already I was trying to connect my project to GitHub and keep track of my daily progress, However I had to restart a section of my project which resulted in me having to delete off GitHub. Right now I plan on try to add more to my GitHub while constantly progressing with the project.

Now What?

My goal again for the month of April to have at least the majority of the project done with the goal in May just to wrap up or tweak any final changes. My goal now is to get most of the development done within this month and move forward with the project as quickly as I can.

I will do more research on MongoDB and make sure I don’t come into any errors. I have still looked at the examples of implementing the node.js and express.js into my project. Like with MongoDB I have found examples of each and how to properly implement them into my project.

Student Signature	Luke Kavanagh

Student Name	Luke Kavanagh
Student Number	X17520686
Course	B.S(Honours) in Computing, Specialising in Software Development

Month: April

What?

April I was finally able to put all my academic focus and attention on the final year project. As mentioned in previous journals, the development of the project was on going and had ups and downs throughout the project. Assessments in April came to an end and gave me more time to complete.

My days increased with the project to at least 5 days of working on the project within the last few weeks increasing to 6 within the last few weeks.

So What?

Implementing my project together was very difficult and it required me to do much more research then I previously expected. I originally tried to get the server connected to the main client page. I needed to download and add multiple dependencies to my project environment however I did get some errors within the project.

I continued to try and get a good understanding of MongoDB. I quickly did a practice video connecting a User interface sign in to the database. I continued to do it for the project however I did come to an error that I need to resolve. I also made a decision to change the look of the user interface then I originally intended. It looks more efficient and has a cleaner design and look to the project.

It will also allow for the user to create an event to sell the ticket. I also wanted to add in a feature button to show who is interested in a particular event. Finally for the user interface I want to make the user interface have mobile navigation. If I do have enough time I will be able to add the feature.

Now What?

As of today 2nd of May 2022, the project is almost complete, Although I need to make the few changes it will be completed in time. I need to check and debug the database as the server isn't working and sends back an error. With two weeks to complete it I hope by then it will be complete without any issues. I have already implemented and designed a sign system That will be connected to the database. I will complete it by connecting it to MongoDB and see the data being saved within the system. As mentioned already it will all be completed by the deadline.

Student Signature	Luke Kavanagh

6.3 Invention Disclosure Form